

Система автоматического хаос-тестирования хостового агента

>  

— □ ×

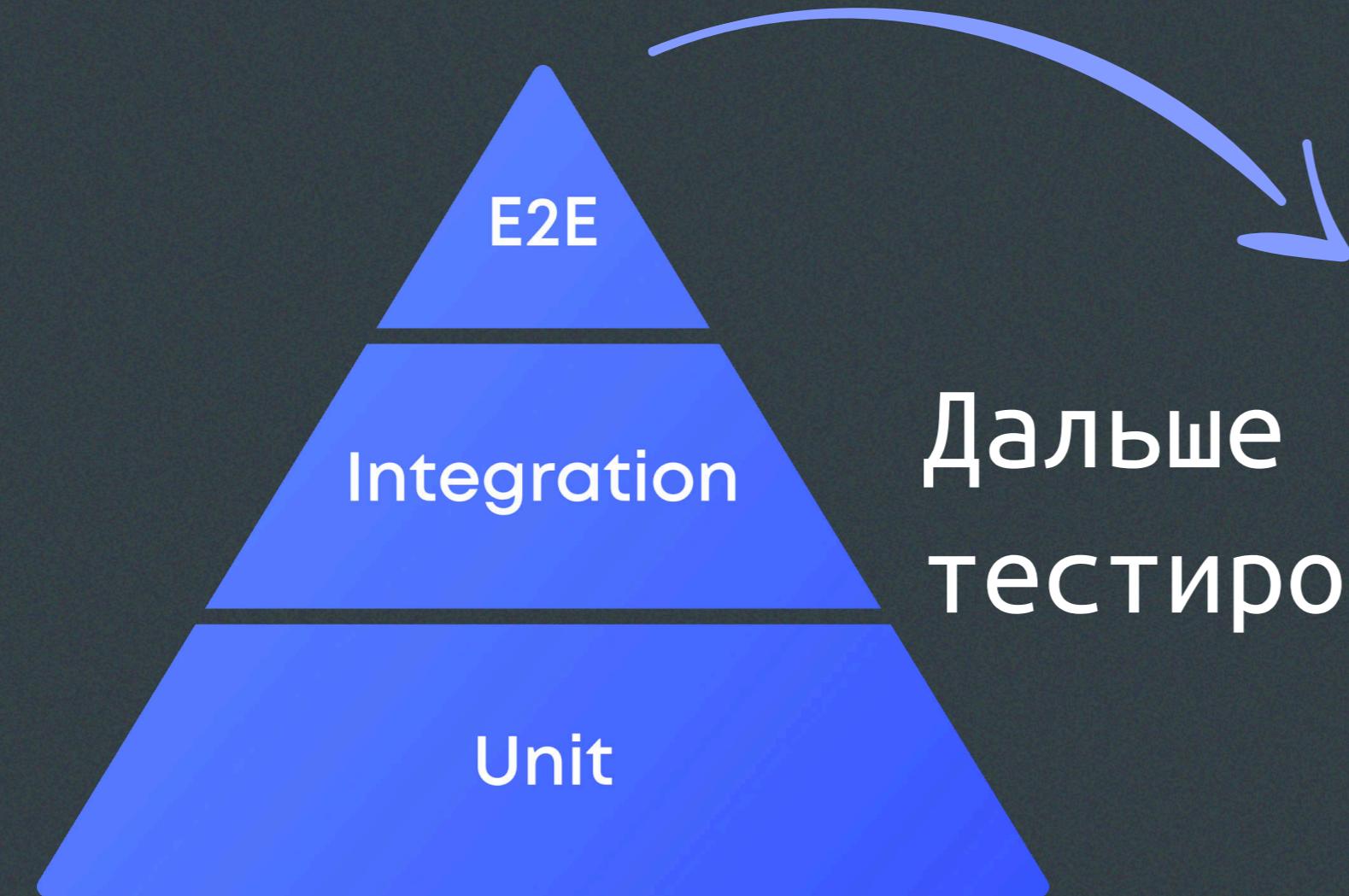
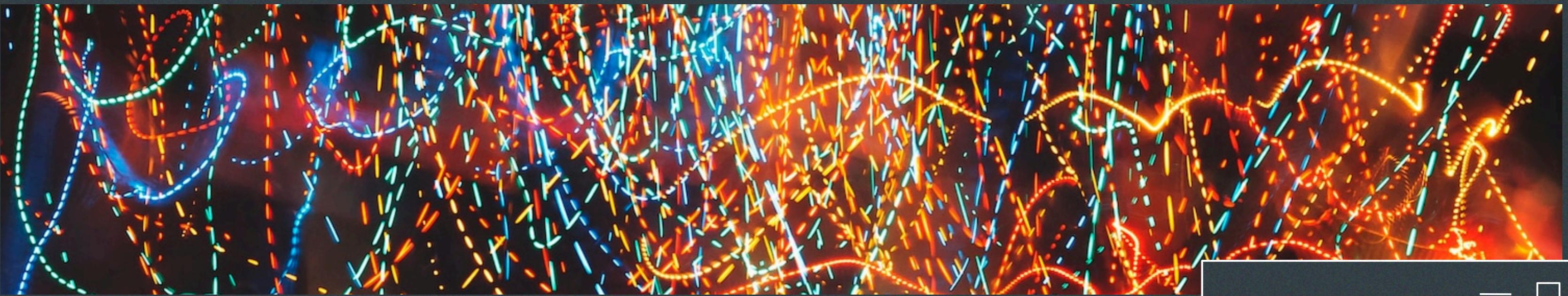
Что это значит?

— □ ×

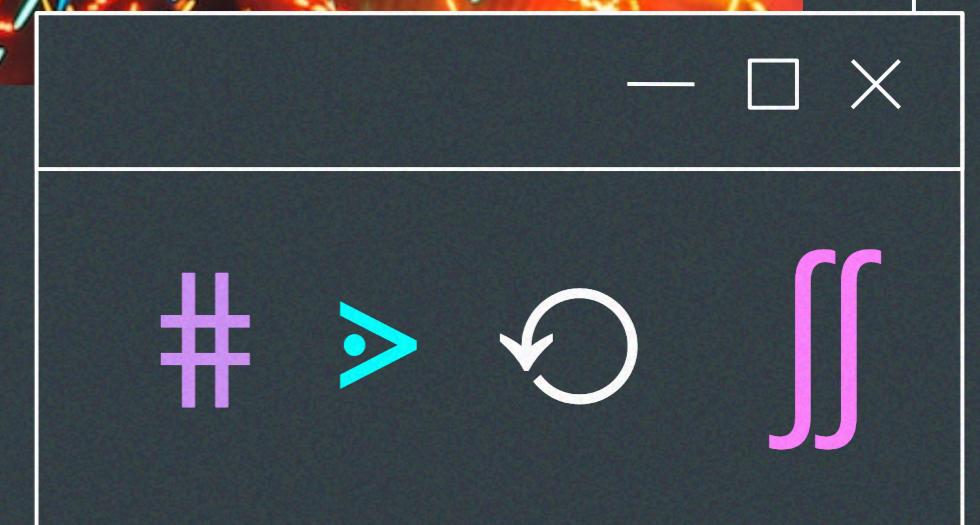
÷ ≥ ↓↑

О ЧЕМ МЫ ГОВОРИМ?

— □ ×



Дальше тоже есть, что
тестировать



— □ ×

Хаос-тестинг ← +



О ЧЕМ МЫ ГОВОРИМ?

- □ ×

Хостовой агент



- □ ×

÷ ≥ ↓↑

Цель

- □ ×

Цель: получить систему автоматизированного хаос-тестирования хостового агента, которая позволит создавать контролируемые условия сбоев и ограничений в среде исполнения агента и предоставит инструменты для мониторинга и анализа его поведения при возникновении этих сбоев.

- □ ×

÷ ≥ ↓↑

— □ ×

Почему это актуально?

— □ ×

÷ ≥ ↓↑

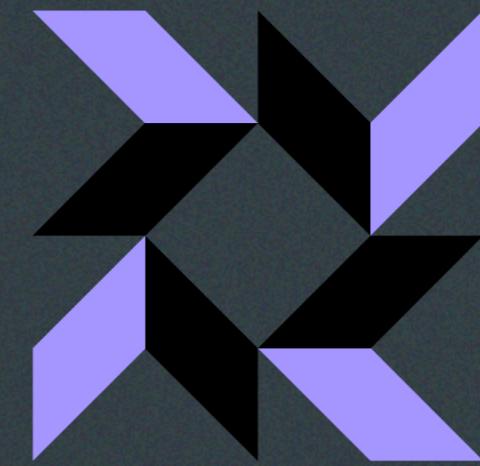
Почему это актуально?

— □ ×

Агенты повсюду



Prometheus



Osquery



DataDog



÷ ≥ ↓↑

— □ ×

Почему это актуально?

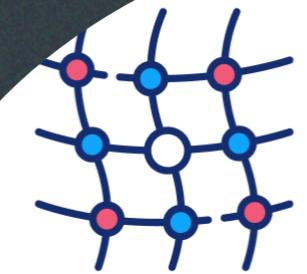
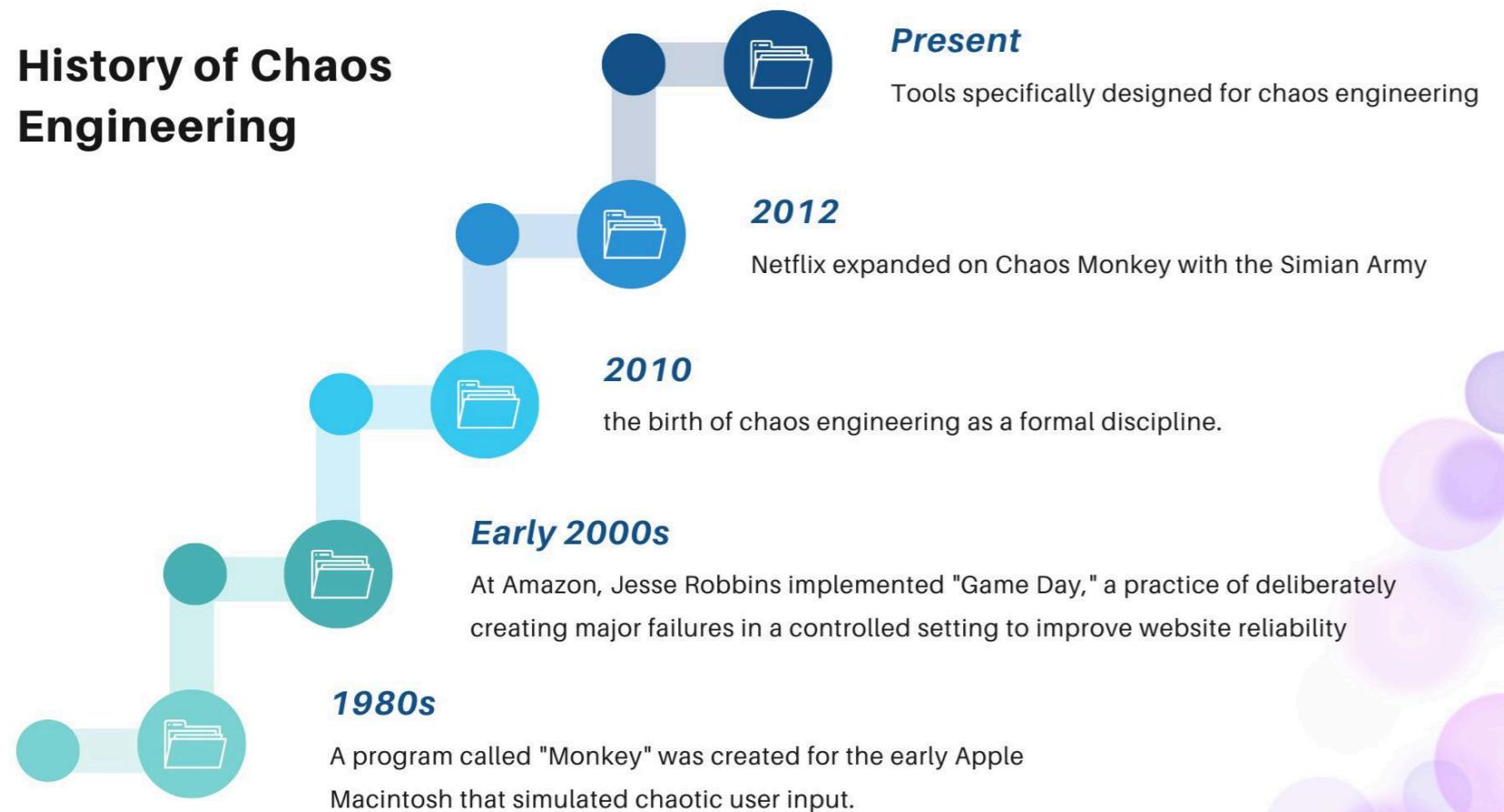
— □ ×



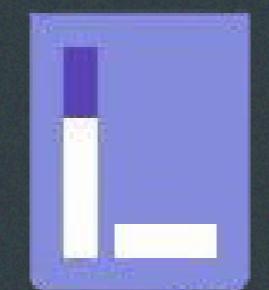
Gremlin



History of Chaos Engineering



Chaos Mesh



Litmus

÷ ≥ ↓↑

— □ ×

Почему это актуально
вместе?

÷ ≥ ↓↑

Почему это актуально вместе?

— □ ×

Распределенная система vs Хостовой агент



÷ ≥ ↓↑

— □ ×

Почему это актуально вместе?

— □ ×

Принципы хаос-инжиниринга: не все хочется применять

1. Гипотеза о стабильном состоянии
2. Максимально близко к реальным сбоям
3. Минимизация радиуса воздействия
4. Автоматизация экспериментов
5. *Мониторинг внешних метрик*

приложения, а не его внутренних проблем

— □ ×

÷ ≥ ↓↑



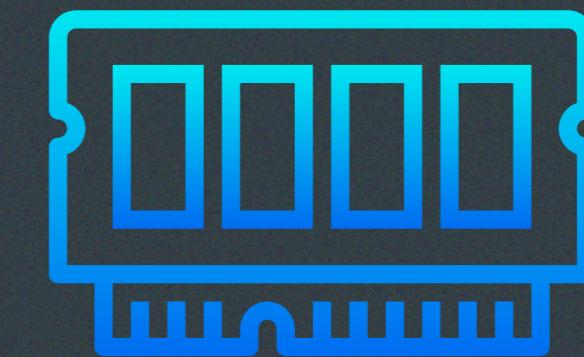
Что требуется от системы?



÷ ≥ ↓↑

Что требуется от системы?

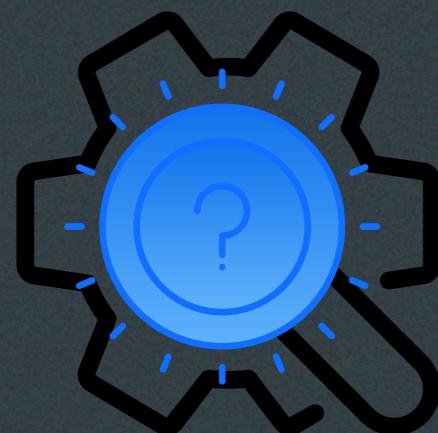
— □ ×



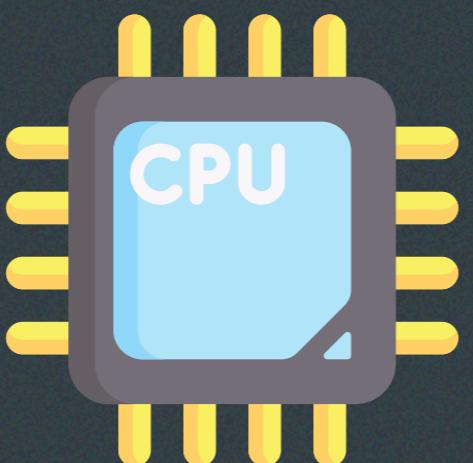
RAM



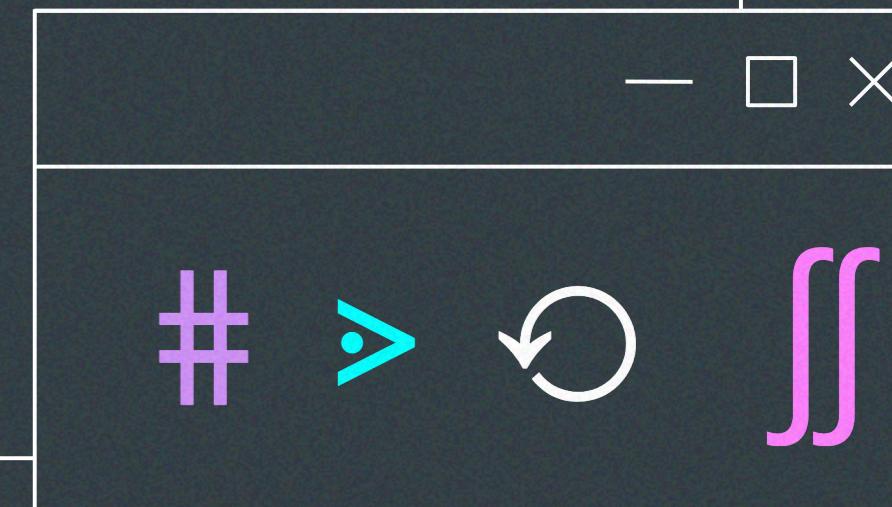
Disk



Utility



Network

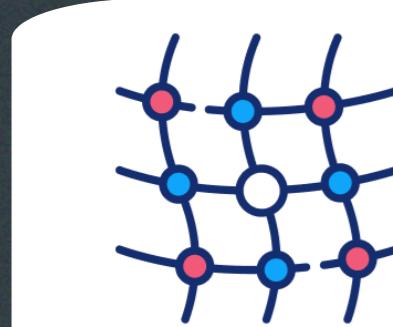


— □ ×

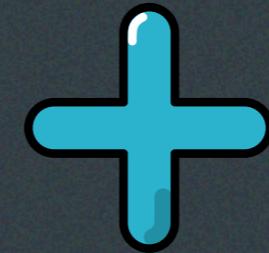
Что в планах?

— □ ×

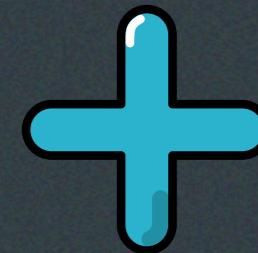
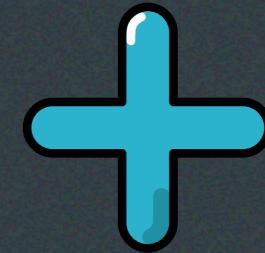
÷ ≥ ↓↑



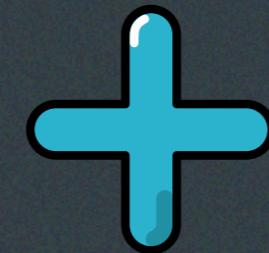
Chaos Mesh



Специфические
сценарии, исходные
которых зависят от
конкретного агента

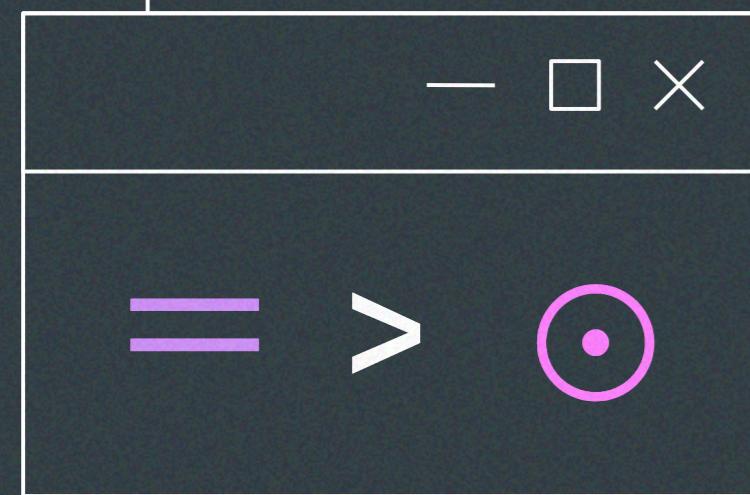


Мониторинг



Вывод подсказки, где
искать проблему, то
есть анализ кода

- □ ×



GitVerse